Serial No.: 09/936,510

IN THE TITLE:

Please amend the title to read:

~~Block Interleave Device, Block Deinterleave Device, Block Interleave Method, and Block Deinterleave Method~~ Block Interleaving Apparatus, Block Deinterleaving Apparatus, Block Interleaving Method and Block Deinterleaving Method.

IN THE SPECIFICATION:

Page 4, line 20 to Page 5, line 2, please amend as follows:

That is, assuming that the n-th address is Ab(n), the number of rows of the storage unit is L, the number of columns is M, b is an integer not less than 0, and x is an arbitrary integer not less than 0 and not larger than b,

$$Ab(n)=(Ab(n-1)+M^{**}(b-x))\bmod(L\times M-1) \quad \ldots(1)$$

Further,

$$REG=(M^{**}(b-x))\bmod(L\times M-1)$$

wherein Ab(0) is 0, and M**(b-x) indicates the (b-x)th power of M. The formula "LxM-1" herein means "(LxM)-1".

Page 7, line 21 to Page 31, line 10, please amend as follows:

A block interleaving apparatus according to ~~Claim~~ aspect 1 of the present invention comprises: a storage means to which (LxM) pieces of addresses are allocated (L,M: integers,

$2 \leq L, M)$; an address generation means for generating addresses for writing and reading blocks, each block having (L×M) pieces of data as a unit to be subjected to block interleaving, in/from the storage means; and a control means for controlling the storage means so that the storage means switches the operation between the data writing and the data reading, by using the addresses generated by the address generation means; and the address generation means comprises: a multiplication means for generating the product of α (α: integer, $2 \leq \alpha$) and $M^{(b-x)}$ (x: integer, $0 \leq x \leq b$, b: integer, $0 \leq b$), every time a block of a block number b is inputted; a first overflow processing means having a first comparison means for comparing the product obtained by the multiplication means with a comparison reference value L×M-1, and subtracting, as much as possible, the L×M-1 from the product on the basis of the result of the comparison to suppress overflow of the product, thereby outputting an address increment value REG corresponding to of the block having the block number b; an addition means for successively adding the (n-1)th (n: integer, $1 \leq n \leq L \times M-1$) address Ab(n-1) of the block having the block number b, to the address increment value REG outputted from the first overflow processing means, every time the block of the block number b

is inputted, thereby successively generating the n-th address Ab(n) in the block of the block number b; and a second overflow processing means having a second comparison means for comparing the sum obtained by the addition means with the comparison reference value L×M-1, and subtracting, as much as possible, the L×M-1 from the sum on the basis of the result of the comparison to suppress overflow of the sum, thereby outputting an address to be actually supplied to the storage means; wherein, when the first comparison means compares the product obtained by the multiplication with the comparison reference value L×M-1, the first comparison means employs, as a comparison reference value instead of the L×M-1, the minimum value A which exceeds the L×M-1 and is included in the product.

In the block interleaving apparatus according to ~~Claim~~ aspect 1 of the present invention, since the above-described address generation is carried out when writing or reading data in/from the storage means, block interleaving operation on a single plane of the storage means having a storage area of one block is realized, and the circuit scale of the address generation means is reduced.

A block interleaving apparatus according to ~~Claim~~ aspect 2 of the present invention comprises: a storage means to which (L×M) pieces of addresses are allocated (L,M: integers, 2≤L,M); an address generation means for generating addresses for writing and reading blocks, each block having (L×M) pieces of data as a unit to be subjected to block interleaving, in/from the storage means; and a control means for controlling the storage means so that the storage means switches the operation between the data writing and the data reading, by using the addresses generated by the address generation means; and the address generation means includes: an address increment value storage means for storing an address increment value REG(b) corresponding to a block having a block number b (b: integer, 1≤b); a first initial value setting means for setting α (α: integer, 2≤α) as an address increment value REG(0) corresponding to a block having a block number 0, in the address increment value storage means; a multiplication means for multiplying the output value REG(c) (c=b-1) from the address increment value storage means by M; a first overflow processing means having a first comparison means for comparing the product obtained by the multiplication means with a comparison reference value L×M-1, and subtracting, as much as

possible, the L×M-1 from the product on the basis of the comparison result to perform a calculation equivalent to "α×M**(b-x)mod(L×M-1)" (M**(b-x) means $M^{(b-x)}$, mod is the remainder, x is an integer, 0≤x≤b), thereby suppressing overflow, and outputting the calculation result as an address increment value REG(b) corresponding to the block of the block number b to the address increment value storage means; an address storage means for storing the n-th (n: integer, 1≤n≤L×M-1) address Ab(n) in the block of the block number b (b: integer, 1≤b), and outputting it to an address input terminal of the storage means; a second initial value setting means for setting the 0th address Ab(0) corresponding to the block of the block number b in the address storage means; an addition means for adding the address increment value REG(b) from the address increment value storage means, to the output value Ab(p) (p=n-1) from the address storage means; and a second overflow processing means having a second comparison means for comparing the sum obtained by the addition means with the comparison reference value L×M-1, and subtracting, as much as possible, the L×M-1 from the sum on the basis of the comparison result to perform a calculation equivalent to "(Ab(n-1)+α×M**(b-x))mod(L×M-1)", thereby suppressing overflow

of the sum, and outputting the calculation result as the n-th address Ab(n) of the block having the block number b to the address storage means; wherein, when the first comparison means compares the product obtained by the multiplication with the comparison reference value L×M-1, the first comparison means employs, as a comparison reference value instead of the L×M-1, the minimum value A which exceeds the L×M-1 and is included in the product.

In the block interleaving apparatus according to ~~Claim~~ aspect 2 of the present invention, since the above-described address generation is carried out when writing or reading data in/from the storage means, block interleaving operation on a single plane of the storage means having a storage area of one block is realized, and the circuit scale of the address generation means is reduced.

According to ~~Claim~~ aspect 3 of the present invention, in the block interleaving apparatus of ~~Claim~~ aspect 2, the first initial value setting means comprises: a first constant generation means for generating the α; and a first selector for selecting the α from the first constant generation means when a reset signal is inputted, and outputting it to the address increment value storage means; and the first overflow

processing means comprises: a second selector for receiving the output of the multiplication means and the output of the address increment value storage means, and selecting the output of the multiplication means at the beginning of each block, and selecting the output of the address increment value storage means during a period of time other than the beginning of the block; a first comparison means for comparing the output of the second selector with the comparison reference value A; first subtraction means for subtracting the L×M-1 from the output of the second selector; and a third selector for receiving the output of the second selector and the output of the first subtraction means, and selecting the output of the first subtraction means when the output of the second selector is equal to or larger than the comparison reference value, and selecting the output of the second selector when the output of the second selector is smaller than the comparison reference value; wherein the output of the third selector is supplied to the address increment value storage means through the first selector during a period of time when the reset signal is not inputted.

In the block interleaving apparatus according to ~~Claim~~ aspect 3 of the present invention, since the first initial

value setting means and the first overflow processing means are constructed as described above, a remainder is obtained immediately at a point of time where the remainder can be obtained and then multiplication by M is carried out, whereby the remainder is obtained by power multiplication of the value of M equivalently. Therefore, multiplication and remainder calculation do not take much time, and address generation is realized even by low-speed arithmetic processing.

According to ~~Claim~~ aspect 4 of the present invention, in the block interleaving apparatus of ~~Claim~~ aspect 2, the first comparison means employs, as a comparison reference value instead of the minimum value A exceeding the L×M-1, a value B which satisfies L×M-1<B<A and is selected so that the number of logic gates constituting the comparison means is minimized.

In the block interleaving apparatus according to ~~Claim~~ aspect 4 of the present invention, since the above-described comparison reference value is employed, the circuit area of the first comparison means is further reduced, whereby the circuit scale of the address generation means is further reduced.

According to ~~Claim~~ aspect 5 of the present invention, in the block interleaving apparatus of ~~Claim~~ aspect 2, the second

initial value setting means comprises: a second constant generation means for generating a value 0; and a fourth selector for selecting the value 0 from the second constant generation means when a reset signal is inputted, and outputting it to the address storage means; and the second overflow processing means comprises: a second comparison means for comparing the output of the addition means with the comparison reference value L×M-1; a second subtraction means for subtracting the comparison reference value L×M-1 from the output of the addition means; and a fifth selector for receiving the output of the addition means and the output of the second subtraction means, and selecting the output of the second subtraction means when the output of the addition means is equal to or larger than the comparison reference value, and selecting the output of the addition means when the output of the addition means is smaller than the comparison reference value; wherein the output of the fifth selector is supplied to the address storage means through the fourth selector during a period of time when the reset signal is not inputted.

Since the block interleaving apparatus according to ~~Claim~~ aspect 5 of the present invention is constructed as described above, the construction of the second overflow processing

means is simplified as compared with that of the first overflow processing means, whereby the circuit scale of the address generation means is further reduced.

According to ~~Claim~~ aspect 6 of the present invention, in the block interleaving apparatus of ~~Claim~~ aspect 2, the values of α and $L \times M-1$ are set so that no common divisor exists between them.

Since the block interleaving apparatus according to ~~Claim~~ aspect 6 of the present invention is constructed as described above, the address generation rule is prevented from failing, and the storage means and the address generation means are optimized, whereby block interleaving is achieved with the minimum circuit scale.

According to ~~Claim~~ aspect 7 of the present invention, in the block interleaving apparatus of ~~Claim~~ aspect 2, the values of α and $M^{(-x)}$ are set so that α is not equal to $M^{(-x)}$.

Since the block interleaving apparatus according to ~~Claim~~ aspect 7 of the present invention is constructed as described above, continuous writing of addresses is prevented at the time of initial writing, and the storage means and the address generation means are optimized, whereby block interleaving is achieved with the minimum circuit scale.

According to ~~Claim~~ aspect 8 of the present invention, in the block interleaving apparatus of ~~Claim~~ aspect 2, the values of α, L, and M are set at 20, 8, and 203, respectively.

Since the block interleaving apparatus according to ~~Claim~~ aspect 8 of the present invention is constituted as described above, the circuit area of the first comparison means as a component of the address generation means is reduced, and the storage means and the address generation means are optimized, whereby block interleaving is achieved with the minimum circuit scale.

According to ~~Claim~~ aspect 9 of the present invention, in the block interleaving apparatus of ~~Claim~~ aspect 2, the values of (L,M) are set at any of 72 possible values as follows: L=96×X (X=1,2,4), M=2,...,13; or M=2,...,13, L=96×X (X=1,2,4).

Since the block interleaving apparatus according to ~~Claim~~ aspect 9 is constructed as described above, the circuit area of the first comparison means as a component of the address generation means is reduced, and the storage means and the address generation means are optimized, whereby block interleaving is achieved with the minimum circuit scale.

A block deinterleaving apparatus according to ~~Claim~~ aspect 10 of the present invention comprises: a storage means

to which (L×M) pieces of addresses are allocated (L,M: integers, $2 \leq L,M$); an address generation means for generating addresses for writing and reading blocks, each block having (L×M) pieces of data as a unit to be subjected to block interleaving, in/from the storage means; and a control means for controlling the storage means so that the storage means switches the operation between the data writing and the data reading, by using the addresses generated by the address generation means; and the address generation means comprises: a multiplication means for generating the product of α (α: integer, $2 \leq \alpha$) and $L^{(b-x)}$ (x: integer, $0 \leq x \leq b$, b: integer, $0 \leq b$), every time a block of a block number b is inputted; a first overflow processing means having a first comparison means for comparing the product obtained by the multiplication means with a comparison reference value L×M-1, and subtracting, as much as possible, the L×M-1 from the product on the basis of the comparison result to suppress overflow of the product, thereby outputting an address increment value REG corresponding to the block having the block number b; an addition means for successively adding the (n-1)th (n: integer, $1 \leq n \leq L \times M-1$) address Ab(n-1) of the block having the block number b, to the address increment value REG outputted

from the first overflow processing means, every time the block of the block number b is inputted, thereby successively generating the n-th address Ab(n) in the block of the block number b; and a second overflow processing means having a second comparison means for comparing the sum obtained by the addition means with the comparison reference value L×M-1, and subtracting, as much as possible, the L×M-1 from the sum on the basis of the comparison result to suppress overflow of the sum, thereby outputting an address to be actually supplied to the storage means; wherein, when the first comparison means compares the product obtained by the multiplication with the comparison reference value L×M-1, the first comparison means employs, as a comparison reference value instead of the L×M-1, the minimum value A which exceeds the L×M-1 and is included in the product.

In the block deinterleaving apparatus according to ~~Claim~~ aspect 10 of the present invention, since the above-described address generation is carried out when writing or reading data in/from the storage means, block deinterleaving on a single plane of the storage means having a storage area of one block is realized, and the circuit scale of the address generation means is reduced.

A block deinterleaving apparatus according to ~~Claim~~ aspect 11 of the present invention comprises: a storage means to which (L×M) pieces of addresses are allocated (L,M: integers, $2 \leq L,M$); an address generation means for generating addresses for writing and reading blocks, each block having (L×M) pieces of data as a unit to be subjected to block interleaving, in/from the storage means; and a control means for controlling the storage means so that the storage means switches the operation between the data writing and the data reading, by using the addresses generated by the address generation means; and the address generation means includes: an address increment value storage means for storing an address increment value REG(b) corresponding to a block having a block number b (b: integer, $1 \leq b$); a first initial value setting means for setting α (α: integer, $2 \leq \alpha$) as an address increment value REG(0) corresponding to a block having a block number 0, in the address increment value storage means; a multiplication means for multiplying the output value REG(c) (c=b-1) from the address increment value storage means by L; a first overflow processing means having a first comparison means for comparing the product obtained by the multiplication means with a comparison reference value L×M-1, and

subtracting, as much as possible, the L×M-1 from the product on the basis of the comparison result to perform a calculation equivalent to "α×L**(b-x)mod(L×M-1)" (L**(b-x) indicates $L^{(b-x)}$, mod is the remainder, x is an integer, 0≤x≤b), thereby suppressing overflow, and outputting the calculation result as an address increment value REG(b) corresponding to the block of the block number b to the address increment value storage means; an address storage means for storing the n-th (n: integer, 1≤n≤L×M-1) address Ab(n) in the block of the block number b, and outputting it to an address input terminal of the storage means; a second initial value setting means for setting the 0th address Ab(0) of the block of the block number b in the address storage means; an addition means for adding the address increment value REG(b) from the address increment value storage means to the output value Ab(p) (p=n-1) from the address storage means; a second overflow processing means having a second comparison means for comparing the sum obtained by the addition means with the comparison reference value L×M-1, and subtracting, as much as possible, the L×M-1 from the sum on the basis of the comparison result to perform a calculation equivalent to "(Ab(n-1)+α×L**(b-x))mod(L×M-1)", thereby suppressing overflow of the sum, and outputting the

calculation result as the n-th address Ab(n) corresponding to the block having the block number b to the address storage means; wherein, when the first comparison means compares the product from the multiplication means with the comparison reference value L×M-1, the first comparison means employs, as a comparison reference value instead of the L×M-1, the minimum value A which exceeds the L×M-1 and is included in the product.

In the block deinterleaving apparatus according to ~~Claim~~ aspect 11 of the present invention, since the above-described address generation is carried out when writing or reading data in/from the storage means, block deinterleaving on a single plane of the storage means having a storage area of one block is realized, and the circuit scale of the address generation means is reduced.

According to ~~Claim~~ aspect 12 of the present invention, in the block deinterleaving apparatus of ~~Claim~~ aspect 11, the first initial value setting means comprises: a first constant generation means for generating the α; and a first selector for selecting the α from the first constant generation means when a reset signal is inputted, and outputting it to the address increment value storage means; and the first overflow

processing means comprises: a second selector for receiving the output of the multiplication means and the output of the address increment value storage means, and selecting the output of the multiplication means at the beginning of each block, and selecting the output of the address increment value storage means during a period of time other than the beginning of the block; a first comparison means for comparing the output of the second selector with the comparison reference value A; a first subtraction means for subtracting the L×M-1 from the output of the second selector; and a third selector for receiving the output of the second selector and the output of the first subtraction means, and selecting the output of the first subtraction means when the output of the second selector is equal to or larger than the comparison reference value, and selecting the output of the second selector when the output of the second selector is smaller than the comparison reference value; wherein the output of the third selector is supplied to the address increment value storage means through the first selector during a period of time when the reset signal is not inputted.

In the block deinterleaving apparatus according to ~~Claim~~ aspect 12 of the present invention, since the first initial

value setting means and the first overflow processing means are constructed as described above, a remainder is obtained immediately at a point of time where the remainder can be obtained and then multiplication by M is performed, whereby the remainder is obtained by power multiplication of the value of M equivalently. Therefore, multiplication and remainder calculation do not take much time, and address generation is realized even by low-speed arithmetic processing.

According to ~~Claim~~ aspect 13 of the present invention, in the block deinterleaving apparatus of ~~Claim~~ aspect 11, the first comparison means employs, as a comparison reference value instead of the minimum value A exceeding the L×M-1, a value B which satisfies L×M-1<B<A and is selected so that the number of logic gates constituting the comparison means is minimized.

In the block deinterleaving apparatus according to ~~Claim~~ aspect 13 of the present invention, since the above-described comparison reference value is employed, the circuit area of the first comparison means is further reduced, whereby the circuit scale of the address generation means is further reduced.

According to ~~Claim~~ aspect 14 of the present invention, in the block deinterleaving apparatus of ~~Claim~~ aspect 11, the second initial value setting means comprises: a second constant generation means for generating a value 0; and a fourth selector for selecting the value 0 from the second constant generation means when a reset signal is inputted, and outputting it to the address storage means; and the second overflow processing means comprises: a second comparison means for comparing the output of the addition means with the comparison reference value L×M-1; a second subtraction means for subtracting the comparison reference value L×M-1 from the output of the addition means; and a fifth selector for receiving the output of the addition means and the output of the second subtraction means, and selecting the output of the second subtraction means when the output of the addition means is equal to or larger than the comparison reference value, and selecting the output of the addition means when the output of the addition means is smaller than the comparison reference value; wherein the output of the fifth selector is supplied to the address storage means through the fourth selector during a period of time when the reset signal is not inputted.

Since the block deinterleaving apparatus according to ~~Claim~~ aspect 14 of the present invention is constructed as described above, the construction of the second overflow processing means is simplified as compared with that of the first overflow processing means, whereby the circuit scale of the address generation means is further reduced.

According to ~~Claim~~ aspect 15 of the present invention, in the block deinterleaving apparatus of ~~Claim~~ aspect 11, the values of α and $L \times M-1$ are set so that no common divisor exists between them.

Since the block deinterleaving apparatus according to ~~Claim~~ aspect 15 of the present invention is constructed as described above, the address generation rule is prevented from failing, and the storage means and the address generation means are optimized, whereby block deinterleaving is achieved with the minimum circuit scale.

According to ~~Claim~~ aspect 16 of the present invention, in the block deinterleaving apparatus of ~~Claim~~ aspect 11, the values of α and $L^{(-x)}$ are set so that α is not equal to $L^{(-x)}$.

Since the block deinterleaving apparatus according to ~~Claim~~ aspect 16 of the present invention is constructed as described above, continuous writing of addresses is prevented

at the time of initial writing, and the storage means and the address generation means are optimized, whereby block deinterleaving is achieved with the minimum circuit scale.

According to ~~Claim~~ aspect 17 of the present invention, in the block deinterleaving apparatus of ~~Claim~~ aspect 11, the values of α, L, and M are set at 20, 8, and 203, respectively.

Since the block deinterleaving apparatus according to ~~Claim~~ aspect 17 of the present invention is constructed as described above, the circuit area of the first comparison means as a component of the address generation means is reduced, and the storage means and the address generation means are optimized, whereby block deinterleaving is achieved with the minimum circuit scale.

According to ~~Claim~~ aspect 18 of the present invention, in the block deinterleaving apparatus of ~~Claim~~ aspect 11, the values of (L,M) are set at any of 72 possible values as follows: $L=96\times X$ $(X=1,2,4)$, $M=2,\ldots,13$; or $M=2,\ldots,13$, $L=96\times X$ $(X=1,2,4)$.

Since the block deinterleaving apparatus according to ~~Claim~~ aspect 18 of the present invention is constructed as described above, the circuit area of the first comparison means as a component of the address generation means is

reduced, and the storage means and the address generation means are optimized, whereby block deinterleaving is achieved with the minimum circuit scale.

According to ~~Claim~~ aspect 19 of the present invention, there is provided a block interleaving method for performing block interleaving of data by generating addresses for writing and reading blocks, each block having (L×M) pieces of data (L,M: integers, 2≤L,M) as a unit to be interleaved, in/from a storage means to which (L×M) pieces of addresses are allocated, and controlling the storage means by using the generated addresses so that the storage means switches the operation between the data writing and the data reading: wherein α (integer, 2≤α) is given as an address increment value REG to a block having a block number 0 and, thereafter, the increment value REG is multiplied by M every time the block number increments by 1 and thus obtained REG is used as an address increment value REG of the corresponding block, and when the address increment value REG exceeds L×M-1, the remainder over L×M-1 is used as an increment value instead of the increment value REG to repeat the above-described processing, thereby performing a calculation equivalent to "α×M**(b-x)mod(L×M-1)" (M**(b-x) indicates $M^{(b-x)}$, mod is the

remainder, and x is an integer, $0 \leq x \leq b$) to obtain an address increment value of each block; in the case where Ab(0) is set as an initial value of address in each block and, thereafter, the address increment value REG in this block is successively summed to generate addresses Ab(1) to Ab(n) (n: integer, $1 \leq n \leq L \times M-1$) in this block, when the address exceeds L×M-1, the remainder over L×M-1 is used as an address instead of the address to repeat the above-described processing, thereby generating addresses in each block; and when calculating the address increment value, decision as to whether the remainder is to be obtained or not is made by comparing the address increment value with the L×M-1 using first comparison means and, at this time, the minimum value A which exceeds the L×M-1 and is included in the result of multiplication is used as a comparison reference value instead of the L×M-1.

In the block interleaving method according to ~~Claim~~ aspect 19 of the present invention, since the above-described address generation is performed when writing or reading data in/from the storage means, block interleaving on a single plane of the storage means having a storage area of one block is realized, and the circuit scale of the address generation means is reduced.

According to ~~Claim~~ aspect 20 of the present invention, in the block interleaving method of ~~Claim~~ aspect 19, the first comparison means employs, as a comparison reference value instead of the minimum value A exceeding the L×M-1, a value B which satisfies L×M-1<B<A and is selected so that the number of logic gates constituting the comparison means is minimized.

In the block interleaving method according to ~~Claim~~ aspect 20 of the present invention, since the above-described comparison reference value is employed, the circuit area of the first comparison means is further reduced, whereby the circuit scale of the address generation means is further reduced.

According to ~~Claim~~ aspect 21 of the present invention, in the block interleaving method of ~~Claim~~ aspect 19, the values of α and L×M-1 are set so that no common divisor exists between them.

Since the block interleaving method of ~~Claim~~ aspect 21 is constructed as described above, the address generation rule is prevented from failing, and the storage means and the address generation means are optimized, whereby block interleaving is achieved with the minimum circuit scale.

According to ~~Claim~~ aspect 22 of the present invention, in the block interleaving method of ~~Claim~~ aspect 19, the values of α and $M^{(-x)}$ are set so that α is not equal to $M^{(-x)}$.

Since the block interleaving method of ~~Claim~~ aspect 22 is constructed as described above, continuous writing of addresses is prevented at the time of initial writing, and the storage means and the address generation means are optimized, whereby block interleaving is achieved with the minimum circuit scale.

According to ~~Claim~~ aspect 23 of the present invention, in the block interleaving method of ~~Claim~~ aspect 19, the values of α, L, and M are set at 20, 8, and 203, respectively.

Since the block interleaving method of ~~Claim~~ aspect 23 is constructed as described above, the circuit area of the first comparison means as a component of the address generation means is reduced, and the storage means and the address generation means are optimized, whereby block interleaving is achieved with the minimum circuit scale.

According to ~~Claim~~ aspect 24 of the present invention, in the block interleaving method of ~~Claim~~ aspect 19, the values of (L,M) are set at any of 72 possible values as follows: $L=96\times X$ $(X=1,2,4)$, $M=2,\ldots,13$; or $M=2,\ldots,13$, $L=96\times X$ $(X=1,2,4)$.

Since the block interleaving method according to ~~Claim~~ aspect 24 is constituted as described above, the circuit area of the first comparison means as a component of the address generation means is reduced, and the storage means and the address generation means are optimized, whereby block interleaving is achieved with the minimum circuit scale.

According to ~~Claim~~ aspect 25 of the present invention, there is provided a block deinterleaving method for performing block deinterleaving of data by generating addresses for writing and reading blocks, each block having (L×M) pieces of data (L,M: integers, $2 \leq L,M$) as a unit to be deinterleaved, in/from storage means to which (L×M) pieces of addresses are allocated, and controlling the storage means by using the generated addresses so that the storage means switches the operation between writing and reading of the data: wherein, α (integer, $2 \leq \alpha$) is given as an address increment value REG to a block having a block number 0 and, thereafter, the increment value REG is multiplied by L every time the block number increments by 1 and thus obtained REG is used as an address increment value REG of the corresponding block, and when the address increment value REG exceeds L×M-1, the remainder over L×M-1 is used as an increment value instead of the increment

value REG to repeat the above-described processing, thereby performing a calculation equivalent to "$\alpha \times L^{**}(b-x) \bmod (L \times M-1)$" ($L^{**}(b-x)$ indicates $L^{(b-x)}$, mod is the remainder, and x is an integer, $0 \leq x \leq b$) to obtain an address increment value of each block; in the case where Ab(0) is set as an initial value of address in each block and, thereafter, the address increment value REG in this block is successively summed to generate addresses Ab(1) to Ab(n) (n: integer, $1 \leq n \leq L \times M-1$) in this block, when the address exceeds L×M-1, the remainder over L×M-1 is used as an address instead of the address to repeat the above-described processing, thereby generating addresses in each block; and when calculating the address increment value, decision as to whether the remainder is to be obtained or not is made by comparing the address increment value with the L×M-1 using first comparison means and, at this time, the minimum value A which exceeds the L×M-1 and is included in the result of multiplication is used as a comparison reference value instead of the L×M-1.

In the block deinterleaving method according to ~~Claim~~ aspect 25 of the present invention, since the above-described address generation is performed when writing or reading data in/from the storage means, block deinterleaving on a single

plane of the storage means having a storage area of one block is realized, and the circuit scale of the address generation means is reduced.

According to ~~Claim~~ aspect 26 of the present invention, in the block deinterleaving method of ~~Claim~~ aspect 25, the first comparison means employs, as a comparison reference value instead of the minimum value A exceeding the L×M-1, a value B which satisfies L×M-1<B<A and is selected so that the number of logic gates constituting the comparison means is minimized.

In the block deinterleaving method according to ~~Claim~~ aspect 26 of the present invention, since the above-described comparison reference value is employed, the circuit area of the first comparison means is reduced, whereby the circuit scale of the address generation means is further reduced.

According to ~~Claim~~ aspect 27 of the present invention, in the block deinterleaving method of ~~Claim~~ aspect 25, the values of α and L×M-1 are set so that no common divisor exists between them.

Since the block deinterleaving method according to ~~Claim~~ aspect 27 of the present invention is constructed as described above, the address generation rule is prevented from failing, and the storage means and the address generation means are

optimized, whereby block deinterleaving is achieved with the minimum circuit scale.

According to ~~Claim~~ aspect 28 of the present invention, in the block deinterleaving method of ~~Claim~~ aspect 25, the values of α and $L^{(-x)}$ are set so that α is not equal to $L^{(-x)}$.

Since the block deinterleaving method of ~~Claim~~ aspect 28 is constructed as described above, continuous writing of addresses is prevented at the time of initial writing, and the storage means and the address generation means are optimized, whereby block deinterleaving is achieved with the minimum circuit scale.

According to ~~Claim~~ aspect 29 of the present invention, in the block deinterleaving method of ~~Claim~~ aspect 25, the values of α, L, and M are set at 20, 8, and 203, respectively.

Since the block deinterleaving method according to ~~Claim~~ aspect 29 of the present invention is constructed as described above, the circuit area of the first comparison means as a component of the address generation means is reduced, and the storage means and the address generation means are optimized, whereby block deinterleaving is achieved with the minimum circuit scale.

According to ~~Claim~~ aspect 30 of the present invention, in the block deinterleaving method of ~~Claim~~ aspect 25, the values of (L,M) are set at any of 72 possible values as follows: L=96×X (X=1,2,4), M=2,...,13; or M=2,...,13, L=96×X (X=1,2,4).

Since the block deinterleaving method according to ~~Claim~~ aspect 30 of the present invention is constructed as described above, the circuit area of the first comparison means as a component of the address generation means is reduced, and the storage means and the address generation means are optimized, whereby block deinterleaving is achieved with the minimum circuit scale.

Page 35, line 15 to Page 37, line 4, please amend as follows:

In the address generation unit 103 shown in figure 1, reference numeral 110 denotes a constant generator for generating a constant M; 113 denotes a register in which an initial value α is set; and 111 denotes a multiplier for multiplying an output signal from a register 113 by the initial value M. The multiplier 111 corresponds to a multiplication means for generating a product of α (α:

integer, $\alpha \geq 2$) and $M^{(b-x)}$ (x,b: integers, $0 \leq x \leq b$, $0 \leq b$) every time a block of block number b is inputted. Reference numeral 140 denotes an overflow processing unit to be used when the output from the multiplier 111 overflows. This overflow processing unit 140 corresponds to a first overflow processing means which has a first comparison means for comparing the product from the multiplication means with a comparison reference value L×M-1, and subtracts, as much as possible, the L×M-1 per clock from the product on the basis of the comparison result to suppress overflow of the product, and outputs an address increment REG of the block having the block number b. Reference numeral 121 denotes a switch (second selector) for selecting either an output signal from the multiplier 111 or an output signal from a ~~selector 124,~~ register 113, according to the NBLOCKSYNC signal supplied from the input terminal 102 as a control signal; 122 denotes a subtracter (first subtracter) for subtracting (L×M-1) from the output signal from the selector 121; 123 denotes a comparator (first comparison means) for comparing the output signal from the selector 121 with (L×M-1); 124 denotes a switch (third selector) for selecting either the output signal from the subtracter 122 or the output signal from the selector 121,

according to the output signal from the comparator 123 as a control signal; 118 denotes a constant generator (first constant generation means) for generating an initial value α; 126 denotes a switch (first selector) for selecting either the output signal from the constant generator 118 or the output signal from the selector 124, according to the NRST signal from the input terminal 114 as a control signal, and outputting it to the register (address increment value storage means) 113; 128 denotes a switch (selector) for selecting either the output signal from the register 113 or the output signal from a register 127, according to the NBLOCKSYNC signal as a control signal; and 127 denotes a register to which the output signal from the selector 128 is inputted.

Page 44, line 18 to Page 45, line 10, please amend as follows:

The selector 121 receives the output of the ~~overflow processing unit 140~~ and the output of the ~~selector 124.~~ register 113. When the input data corresponds to the head of the block, a block head input data sync signal 102 is input, and the selector 121 selects the output of the

multiplier 111. In other cases, the selector 121 selects the output of the ~~selector 124.~~register 113. The output of the selector 121 is compared with L×M-1 by the comparator 123. The selector 124 receives the output of the subtracter 122 which subtracts L×M-1 from the output of the selector 121, and the output of the selector 121. When the comparator 123 decides that the output of the selector 121 is equal to or larger than L×M-1, the selector 124 selects the output of the subtracter 122. In other cases, the selector 124 selects the output of the selector 121. The output of the selector 124 is inputted to the register 113. In this way, when the input to the overflow processing unit 140 exceeds L×M-1, the overflow processing unit 140 repeats subtraction of L×M-1 per clock from the input to keep the value equal to or smaller than L×M-1.

Page 45, line 19 to Page 46, line 5, delete paragraph in its entirety.

~~When the input data to the overflow processing unit 140 exceeds L×M-1, the overflow processing unit 140 repeats subtraction of "L×M-1" by an internal loop until the input~~

~~data becomes equal to or smaller than L×M-1, and outputs the result to the register 113. The output of the register 113 is again multiplied by the output "M" of the constant generator 110 by the multiplier 111, and the product is inputted to the overflow processing unit 140. The above-described operation is repeated until L×M pieces of data are inputted. When L×M pieces of data have been inputted, the register 127 is updated to the output value of the register 113 by the block head input data sync signal 102.~~

Page 48, lines 5-16, please amend as follows:

At time t0, since a signal NBLOCKSYNC supplied from the input terminal 102 is at a high level (= value "1"; hereinafter referred to as "H"), the selector 121 does not select the output of the multiplier 111 but selects the output of the ~~selector 124.~~register 113. Although the output value of the ~~selector 124~~ register 113 is indefinite, when it exceeds L×M-1 (in this example, 4×5-1=19), the selector 124 continues to select the output of the subtracter 122 until this value becomes equal to or smaller than L×M-1. When the output value from the selector 124 is equal to or smaller than

L×M-1 from the beginning, the selector 124 selects the output of the selector 121 and, therefore, the output of the selector 124 becomes an indefinite value not larger than L×M-1.

Page 50, lines 12-22, please amend as follows:

At time t4, the value "10" which was input to the register 113 at time t3 is outputted, and the multiplier 111 multiplies this value "10" by the output value "5" from the constant generator 110. However, the selector 121 does not select the product "50" but selects the output value from the ~~selector 124. Since the output value from the selector 124 has become "10" at time t3 and the selector 124 selects this value "10" from the selector 121, this value "10" is retained in a loop constituted by the selectors 121 and 124.~~ register 113. Further, since the selector 126 selects the output of the selector 124, the value "10" is input to the register 113.

Page 52, lines 10-15, please amend as follows:

At time t24, the register 113 outputs the value "31" while the multiplier 111 outputs the value "155", and the

selector 121 selects the output value "31" from the ~~selector 124.~~ register 113. The selector 124 selects the output value "12" from the subtracter 122 according to the decision of the comparator 123, and the selector 126 inputs this value "12" to the register 113.

Page 52, line 22 to Page 53, line 2, please amend as follows:

At time t25, the register 113 outputs the value "12" while the multiplier 111 outputs the value "60", and the selector 121 selects the output value "12" from the ~~selector 124.~~ register 113. The selector 126 inputs this value "12" to the register 113.

Since the selector 128 inputs the output value "10" from the selector ~~127~~ 128 to the ~~selector~~ register 127, this value "10" is retained.

Page 69, line 18 to Page 71, line 5, please amend as follows:

In the address generation unit 3 shown in figure 7, reference numeral 10 denotes a constant generator for generating a constant L, 13 denotes a register in which an initial value α is set; and 11 denotes a multiplier for multiplying an output signal from a register 13 by the constant L, and this multiplier 11 corresponds to a multiplication means for generating a product of α (α: integer, $2 \leq \alpha$) and $M^{(b-x)}$ (x,b: integers, $0 \leq x \leq b$, $0 \leq b$) every time a block of a block number b is inputted. Reference numeral 40 denotes an overflow processing unit provided for the case where the output from the multiplier 11 overflows, and this overflow processing unit 40 corresponds to a first overflow processing means which has a first comparison means for comparing the product from the multiplication means with a reference value L×M-1, and subtracts, as much as possible, the L×M-1 per clock from the product on the basis of the comparison result to suppress overflow of the product, and outputs an address increment REG of the block having the block number b. Reference numeral 21 denotes a switch (second

selector) for selecting one of an output signal from the multiplier 11 and an output signal from a ~~selector 24,~~ register 13, under control of the NBLOCKSYNC signal supplied from the input terminal 2; 22 denotes a subtracter (first subtracter) for subtracting (L×M-1) from the output signal from the selector 21; 23 denotes a comparator (first comparison means) for comparing the output signal from the selector 21 with (L×M-1); 24 denotes a switch (third selector) for selecting one of the output signal from the subtracter 22 and the output signal from the selector 21, under control of the output signal from the comparator 23; 18 denotes a constant generator (first constant generation means) for generating an initial value α; 26 denotes a switch (first selector) for selecting one of the output signal from the constant generator 18 and the output signal from the selector 24, under control of the NRST signal from the input terminal 14, and outputting it to the register (address increment value storage means) 13; 28 denotes a switch (selector) for selecting one of the output signal from the register 13 and the output signal from a register 27, under control of the NBLOCKSYNC signal; and 27 denotes a register to which the output signal from the selector 28 is inputted.

Page 78, line 19 to Page 79, line 11, please amend as follows:

The selector 21 is given the output of the ~~overflow processing unit 40 (output of the~~ multiplier ~~11)~~ 11 and the output of the ~~selector 24.~~ register 13. When the input data corresponds to the head of the block and the head input data sync signal 2 is inputted, the selector 21 selects the output of the multiplier 11. In other cases, the ~~selector 21~~ register 13 selects the output of the selector 24. The output of the selector 21 is compared with L×M-1 by the comparator 23. The selector 24 receives the output of the subtracter 22 which subtracts L×M-1 from the output of the selector 21, and the output of the selector 21. When the comparator 23 decides that the output of the selector 21 is equal to or larger than L×M-1, the selector 24 selects the output of the subtracter 22. In other cases, the selector 24 selects the output of the selector 21. The output of the selector 24 is inputted to the register 13. In this way, when the input to the overflow processing unit 40 exceeds L×M-1, the overflow processing unit

40 repeats subtraction of L×M-1 per clock from the input to make the input value equal to or smaller than L×M-1.

Page 79, line 20 to Page 80, line 5, please delete the paragraph in its entirety.

~~When the input data to the overflow processing unit 40 exceeds L×M-1, the overflow processing unit 40 repeats subtraction of "L×M-1" by an internal loop until the input data becomes equal to or smaller than L×M-1, and outputs the result to the register 13. The output of the register 13 is again multiplied by the output "L" of the constant generator 10 by the multiplier 11, and the product is inputted to the overflow processing unit 40. The above-described operation is repeated until L×M pieces of data are inputted. When L×M pieces of data have been input, the register 27 is updated to the output value of the register 13 by the block head input data sync signal 2.~~

Page 82, lines 4-15, please amend as follows:

At time t0, since a signal NBLOCKSYNC supplied from the input terminal 2 is at a high level (= value "1"; hereinafter referred to as "H"), the selector 21 selects not the output of the multiplier 11 but the output of the ~~selector 24.~~register 13. Although the output value of the ~~selector 24~~ register 13 is indefinite, when it exceeds L×M-1 (in this example, 4×5-1=19), the selector 24 continues to select the output of the subtracter 22 until this value becomes equal to or smaller than L×M-1. When the output value from the selector 24 is equal to or smaller than L×M-1 from the beginning, the selector 24 selects the output of the selector 21 and, therefore, the output of the selector 24 becomes an indefinite value not larger than L×M-1.

Page 83, lines 13-21, please amend as follows:

At time t1, a value "2" is outputted from the register 13, and this is multiplied by a constant L (= value "4") from the constant generator 10 by the multiplier 11. However, at time ~~t1,~~ t2, the selector 21 does not select the product "8".

Further, the selector 26 selects the constant α (= value "2") from the constant generator 18, and this is inputted to the register 13. The selector 28 and the selector 30 select the output of the register 27 and the output of the selector 34, respectively, like those at time t0. These states are the same at time t2.

Page 84, lines 11-20, please amend as follows:

At time t4, the value "8" which was input to the register 13 at time t3 is outputted, and the multiplier 11 multiplies this value "8" by the output value "4" from the constant generator 10. However, the selector 21 does not select the product "32" but selects the output value from the ~~selector 24. Since the output value from the selector 24 has become "8" at time t3 and the selector 24 selects this value "8" from the selector 21, this value "8" is retained in a loop constituted by the selectors 21 and 24.~~ register 13. Further, since the selector 26 selects the output of the selector 24, the value "8" is inputted to the register 13.

Page 86, lines 8-13, please amend as follows:

At time t24, the register 13 outputs the value "13" while the multiplier 11 outputs the value "52", and the selector 21 selects the output value "13" from the ~~selector 24.~~ register 13. The selector 24 selects the output value "13" from the selector 21 according to the decision of the comparator 23, and the selector 26 inputs this value "13" to the register 13.

IN THE CLAIMS:

1. (Currently Amended) A block interleaving apparatus comprising:

a storage means for data writing and data reading, to which (LxM) pieces of addresses are allocated wherein L and M are integers ≥ 2 ~~(L,M: integers, 2≤L,M)~~;

an address generation means for generating addresses for writing and reading blocks, each block having (LxM) pieces of data as a unit to be subjected to block interleaving, ~~in/from~~ into/out of the storage means; and

a control means for controlling the storage means so that the storage means ~~switchesswitches~~, ~~the operation between the data writing and the data reading, by using~~ based on the addresses generated by the address generation ~~means;~~means, between data writing and data reading;

said address generation means comprising:

a multiplication means for generating ~~the~~ a product of α, an integer ≥ 2, ~~(α: integer, 2≤α)~~ and $M^{(b-x)}$ ~~(x: integer, 0≤x≤b, b: integer, 0≤b)~~ wherein x is an integer 0≤x≤b, wherein a block number b is an integer ≥ 0, every time a block of a block number b is inputted;

a first overflow processing means having a first comparison means for comparing the product obtained by the multiplication means with a comparison reference value (LxM)-1, and subtracting~~, as much as possible, the~~ to a difference of not less than zero (LxM)-1 from the product on the basis of the result of the comparison to suppress overflow of the product, thereby outputting an address increment value REG corresponding to of the block having the block number b;

an addition means for successively adding the (n-1)th ~~(n: integer, 1≤n≤LxM-1)~~ address ~~Ab(n-1)~~Ab(n-1), wherein n is an integer 1≤n≤ (LxM)-1 of the block having the block number b, to the address increment value REG outputted from the first overflow processing means, every time the block of the block number b is inputted, thereby successively generating the n-th address Ab(n) in the block of the block number b; and

a second overflow processing means having a second comparison means for comparing ~~the~~ a sum obtained by the addition means with the comparison reference value ~~LxM-1,~~ (LxM)-1, and subtracting~~, as much as possible, the~~ to a different of not less than zero (LxM)-1 ~~LxM-1~~ from the sum on the basis of the result of the comparison to suppress overflow

of the sum, thereby outputting an address to be ~~actually~~ supplied to the storage means;

wherein, when the first comparison means is for comparing ~~compares~~ the product obtained by the multiplication with the comparison reference value ~~LxM-1, the first comparison means employs,~~ (LxM)-1, and for employing as a comparison reference value instead of the ~~LxM-1,~~ value (LxM)-1, the minimum value A which exceeds ~~the LxM-1~~ value (LxM)-1 and is included in the product.

2. (Currently Amended) A block interleaving apparatus comprising:

a storage means for data writing and data reading, to which (L×M) pieces of addresses are allocated ~~(L,M: integers, 2≤L,M)~~ wherein L and M are integers ≥2;

an address generation means for generating addresses for writing and reading blocks, each block having (L×M) pieces of data as a unit to be subjected to block interleaving, ~~in/from~~ into/out of the storage means; and

a control means for controlling the storage means so that the storage means ~~switches the operation between the data writing and the data reading, by using~~ switches, based on the

addresses generated by the address generation ~~means;~~ means, between data writing and data reading;

said address generation means ~~including:~~comprising:

an address increment value storage means for storing an address increment value REG(b) corresponding to a block having a block number ~~b (b: integer, 1≤b);~~b, wherein block number b is an integer ≥ 1;

a first initial value setting means for setting α ~~(α: integer, 2≤α)~~ α, an integer ≥ 2, as an address increment value REG(0) corresponding to a block having a block number 0, in the address increment value storage means;

a multiplication means for multiplying ~~the~~ an output value ~~REG(c) (c=b-1)~~ REG(c), where c=b-1, from the address increment value storage means by M;

a first overflow processing means having a first comparison means for comparing ~~the~~ a product obtained by the multiplication means with a comparison reference value ~~L×M-1,~~ (LxM)-1, and ~~subtracting, as much as possible, the L×M-1~~ subtracting to a difference of not less than zero (LxM)-1 from the product on the basis of the comparison result to perform a calculation equivalent to ~~"α×M**(b-x)mod(L×M-1)" (M**(b-x)~~ αxM**(b-x)mod((LxM)-1), where M**(b-x) means $M^{(b-x)}$, mod is the

remainder, and x is an ~~integer, 0≤x≤b),~~ integer wherein 0≤x≤b, thereby suppressing overflow, and outputting the calculation result as an address increment value REG(b) corresponding to the block of the block number b to the address increment value storage means;

an address storage means for storing the n-th ~~(n: integer, 1≤n≤L×M-1)~~ address Ab(n) in the block of the block number ~~b (b: integer, 1≤b),~~ b, wherein block number b is an integer and b ≥ 1, and n is an integer such that 1 ≤ n ≤ (LxM)-1, and outputting it to an address input terminal of the storage means;

a second initial value setting means for setting the 0th address Ab(0) corresponding to the block of the block number b in the address storage means;

an addition means for adding the address increment value REG(b) from the address increment value storage means, to the output value ~~Ab(p) (p=n-1)~~ Ab(p), where p=n-1 is from the address storage means;

a second overflow processing means having a second comparison means for comparing ~~the~~ a sum obtained by the addition means with the comparison reference value ~~L×M-1,~~ (LxM)-1, and ~~subtracting, as much as possible,~~ subtracting to

a difference of not less than zero the ~~L×M-1~~ (LxM)-1 from the sum on the basis of the comparison result to perform a calculation equivalent to ~~"(Ab(n-1)+α×M**(b-x))mod(L×M-1)",~~ (Ab(n-1)+αxM**(b-x))mod((LxM)-1), thereby suppressing overflow of the sum, and outputting the calculation result as the n-th address Ab(n) of the block having the block number b to the address storage means;

wherein, when the first comparison means ~~compares~~ is for comparing the product obtained by the multiplication with the comparison reference value ~~L×M-1, the first comparison means employs,~~ (LxM)-1, and for employing as a comparison reference value instead of the ~~L×M-1,~~ value (LxM)-1, the minimum value A which exceeds the ~~L×M-1~~ value (LxM)-1 and is included in the product.

3. (Currently Amended) The block interleaving apparatus of Claim ~~2 wherein,~~ 2, wherein

said first initial value setting means comprises:

a first constant generation means for generating the α; and

a first selector for selecting the integer α from the first constant generation means when a reset signal is

inputted, and outputting it to the address increment value storage means;

said first overflow processing means comprises:

a second selector for receiving the output of the multiplication means and ~~the~~ output of the address increment value storage means, and selecting the output of the multiplication means at the beginning of each block, and selecting the output of the address increment value storage means during a period of time other than the beginning of the block;

a first comparison means for comparing ~~the~~ output of the second selector with the comparison reference value A;

first subtraction means for subtracting the ~~LxM-1~~ (LxM)-1 from the output of the second selector; and

a third selector for receiving the output of the second selector and ~~the~~ output of the first subtraction means, and selecting the output of the first subtraction means when the output of the second selector is equal to or larger than the comparison reference value, and selecting the output of the second selector when the output of the second selector is smaller than the comparison reference value;

wherein ~~the~~ output of the third selector is supplied to the address increment value storage means through the first selector during a period of time when the reset signal is not inputted.

4. (Currently Amended) The block interleaving apparatus of Claim ~~2~~ 2, wherein said first comparison means employs, as a comparison reference value instead of the minimum value A exceeding the ~~L×M-1,~~ (LxM)-1, a value B which satisfies ~~L×M-1<B<A~~ ((LxM)-1)<B<A and ~~it~~ is selected so that the number of logic gates constituting the comparison means is minimized.

5. (Currently Amended) The block interleaving apparatus of Claim ~~2 wherein,~~ 2, wherein

said second initial value setting means comprises:

a second constant generation means for generating a value 0; and

a fourth selector for selecting the value 0 from the second constant generation means when a reset signal is inputted, and outputting it to the address storage means;

said second overflow processing means comprises:

a second comparison means for comparing ~~the~~ output of the addition means with the comparison reference value ~~L×M-1;~~ (LxM)-1;

a second subtraction means for subtracting the comparison reference value ~~L×M-1~~ (LxM)-1 from the output of the addition means; and

a fifth selector for receiving the output of the addition means and the output of the second subtraction means, and selecting the output of the second subtraction means when the output of the addition means is equal to or larger than the comparison reference value, and selecting the output of the addition means when the output of the addition means is smaller than the comparison reference value;

wherein ~~the~~ output of the fifth selector is supplied to the address storage means through the fourth selector during a period of time when the reset signal is not inputted.

6. (Currently Amended) The block interleaving apparatus of Claim ~~2~~ 2, wherein the values of α and ~~L×M-1~~ (LxM)-1 are set so that no common divisor exists between them.

7. (Currently Amended) The block interleaving apparatus of Claim 2 ~~wherein~~ wherein, the values of α and $M^{(-x)}$ are set so that α is not equal to $M^{(-x)}$.

8. (Currently Amended) The block interleaving apparatus of Claim 2 ~~wherein~~ wherein, the values of α, L, and M are set at 20, 8, and 203, respectively.

9. (Currently Amended) The block interleaving apparatus of Claim 2 ~~wherein~~ wherein, the values of (L,M) are set at any of 72 possible values as follows:

~~L=96×X (X=1,2,4),~~ L=96×X, where X=1,2,4, and M=2,...,13,

or

M=2,...,13, and L=96×X, where ~~(X=1,2,4)~~ X=1,2,4.

10. (Currently Amended) A block deinterleaving apparatus comprising:

a storage means for data writing and data reading to which (L×M) pieces of addresses are allocated ~~(L,M: integers, 2≤L,M)~~ wherein L and M are integers ≥ 2;

an address generation means for generating addresses for writing and reading blocks, each block having (L×M) pieces of

data as a unit to be subjected to block interleaving, ~~in/from~~ into/out of the storage means; and

a control means for controlling the storage means so that the storage means ~~switches the operation between the data writing and the data reading, by using~~ switches, based on the addresses generated by the address generation ~~means;~~ means between data writing and data reading;

said address generation means comprising:

a multiplication means for generating ~~the~~ a product of ~~α (α: integer, 2≤α)~~ α, an integer ≥2, and $L^{(b-x)}$ ~~(x: integer, 0≤x≤b, b: integer, 0≤b),~~ wherein x is an integer 0≤x≤b, wherein a block number b is an integer ≥0, every time a block of a block number b is inputted;

a first overflow processing means having a first comparison means for comparing the product obtained by the multiplication means with a comparison reference value ~~L×M-1,~~ (LxM)-1, and ~~subtracting, as much as possible, the L×M-1~~ subtracting to a difference of not less than zero (LxM)-1 from the product on the basis of the comparison result to suppress overflow of the product, thereby outputting an address increment value REG corresponding to the block having the block number b;

an addition means for successively adding the (n-1)th ~~(n: integer, 1≤n≤L×M-1)~~ address Ab(n-1), wherein n is an integer 1≤n≤(LxM)-1 of the block having the block number b, to the address increment value REG outputted from the first overflow processing means, every time the block of the block number b is inputted, thereby successively generating the n-th address Ab(n) in the block of the block number b; and

a second overflow processing means having a second comparison means for comparing ~~the~~ a sum obtained by the addition means with the comparison reference value ~~L×M-1,~~ (LxM)-1, and ~~subtracting, as much as possible, the L×M-1~~ subtracting to a difference of not less than zero the (LxM)-1 from the sum on the basis of the comparison result to suppress overflow of the sum, thereby outputting an address to be actually supplied to the storage means;

wherein, when the first comparison means ~~compares~~ is for comparing the product obtained by the multiplication with the comparison reference value ~~L×M-1, the first comparison means employs,~~ (LxM)-1, and for employing as a comparison reference value instead of the ~~L×M-1,~~ value (LxM)-1, the minimum value A which exceeds ~~the L×M-1~~ value (LxM)-1 and is included in the product.

11. (Currently Amended) A block deinterleaving apparatus comprising:

a storage means for data writing and data reading to which (L×M) pieces of addresses are ~~allocated (L,M: integers, 2≤L,M);~~ allocated, wherein L and M are integers ≥2;

an address generation means for generating addresses for writing and reading blocks, each block having (L×M) pieces of data as a unit to be subjected to block interleaving, ~~in/from~~ into/out of the storage means; and

a control means for controlling the storage means so that the storage means ~~switches the operation between the data writing and the data reading, by using~~ switches, based on the addresses generated by the address generation ~~means;~~ means, between data writing and data reading;

said address generation means ~~including:~~ comprising:

an address increment value storage means for storing an address increment value REG(b) corresponding to a block having a block number ~~b (b: integer, 1≤b);~~ b, wherein block number b is an integer ≥1;

a first initial value setting means for setting α ~~(α: integer, 2≤α)~~ α, an integer ≥2, as an address increment

value REG(0) corresponding to a block having a block number 0, in the address increment value storage means;

a multiplication means for multiplying ~~the~~ an output value ~~REG(c) (c=b-1)~~ REG(c), where c=b-1, from the address increment value storage means by L;

a first overflow processing means having a first comparison means for comparing ~~the~~ a product obtained by the multiplication means with a comparison reference value L×M-1, and ~~subtracting, as much as possible, the L×M-1~~ subtracting to a difference of not less than zero (LxM)-1 from the product on the basis of the comparison result to perform a calculation equivalent to ~~"α×L**(b-x)mod(L×M-1)" (L**(b-x) indicates~~ αxL**(b-x)mod((LxM)-1), where L**(b-x) means $L^{(b-x)}$, mod is the remainder, and x is an ~~integer, 0≤x≤b),~~ integer wherein 0≤x≤b, thereby suppressing overflow, and outputting the calculation result as an address increment value REG(b) corresponding to the block of the block number b to the address increment value storage means;

an address storage means for storing the n-th ~~(n: integer, 1≤n≤L×M-1)~~ address Ab(n) in the block of the block number b, where b and n are integers and 1≤n≤(LxM)-1, and

outputting ~~it~~ said n-th address Ab(n) to an address input terminal of the storage means;

a second initial value setting means for setting the 0th address Ab(0) of the block of the block number b in the address storage means;

an addition means for adding the address increment value REG(b) from the address increment value storage means to the output value ~~Ab(p) (p=n-1)~~ Ab(p), where p=n-1 is from the address storage means;

a second overflow processing means having a second comparison means for comparing ~~the~~ a sum obtained by the addition means with the comparison reference value ~~L×M-1,~~ (LxM)-1, and ~~subtracting, as much as possible, the L×M-1~~ subtracting as much as possible to a difference of not less than zero the value (LxM)-1 from the sum on the basis of the comparison result to perform a calculation equivalent to ~~"(Ab(n-1)+α×L**(b-x))mod(L×M-1)",~~ (Ab(n-1)=αxL**(b-x))mod((LxM)-1), thereby suppressing overflow of the sum, and outputting the calculation result as the n-th address Ab(n) corresponding to the block having the block number b to the address storage means;

wherein, when the first comparison means ~~compares~~ is for comparing the product from the multiplication means with the comparison reference value ~~L×M-1, the first comparison means employs,~~ (LxM)-1, and for employing as a comparison reference value instead of the ~~L×M-1,~~ value (LxM)-1, the minimum value A which exceeds the ~~L×M-1~~ value (LxM)-1 and is included in the product.

12. (Currently Amended) The block deinterleaving apparatus of Claim ~~11 wherein,~~ 11, wherein

said first initial value setting means ~~comprises:~~ is for comprising:

a first constant generation means for generating the integer α; and

a first selector for selecting the integer α from the first constant generation means when a reset signal is inputted, and outputting it to the address increment value storage means;

said first overflow processing means ~~comprises:~~ comprising:

a second selector for receiving the output of the multiplication means and ~~the~~ output of the address increment

value storage means, and selecting the output of the multiplication means at the beginning of each block, and selecting the output of the address increment value storage means during a period of time other than the beginning of the block;

a first comparison means for comparing ~~the~~ output of the second selector with the comparison reference value A;

a first subtraction means for subtracting the ~~L×M-1~~ value (LxM)-1 from the output of the second selector; and

a third selector for receiving the output of the second selector and ~~the~~ output of the first subtraction means, and selecting the output of the first subtraction means when the output of the second selector is equal to or larger than the comparison reference value, and selecting the output of the second selector when the output of the second selector is smaller than the comparison reference value;

wherein ~~the~~ output of the third selector is supplied to the address increment value storage means through the first selector during a period of time when the reset signal is not inputted.

13. (Currently Amended) The block deinterleaving apparatus of Claim ~~11~~ 11, wherein said first comparison means ~~employs,~~ is for employing, as a comparison reference value instead of the minimum value A exceeding the ~~LxM-1,~~ value (LxM)-1, a value B which satisfies ~~LxM-1<B<A~~ (LxM)-1<B<A and is selected so that the number of logic gates constituting the comparison means is minimized.

14. (Currently Amended) The block deinterleaving apparatus of Claim ~~11 wherein,~~ 11, wherein

said second initial value setting means comprises:

a second constant generation means for generating a value 0; and

a fourth selector for selecting the value 0 from the second constant generation means when a reset signal is inputted, and outputting ~~it~~ said value 0 to the address storage means;

said second overflow processing means comprises:

a second comparison means for comparing ~~the~~ output of the addition means with the comparison reference value ~~LxM-1;~~ (LxM)-1;

a second subtraction means for subtracting the comparison reference value ~~L×M-1~~ (LxM)-1 from the output of the addition means; and

a fifth selector for receiving the output of the addition means and the output of the second subtraction means, and selecting the output of the second subtraction means when the output of the addition means is equal to or larger than the comparison reference value, and selecting the output of the addition means when the output of the addition means is smaller than the comparison reference value;

wherein ~~the~~ output of the fifth selector is supplied to the address storage means through the fourth selector during a period of time when the reset signal is not inputted.

15. (Currently Amended) The block deinterleaving apparatus of Claim ~~11~~ 11, wherein the values of α and ~~L×M-1~~ (LxM)-1 are set so that no common divisor exists between them.

16. (Currently Amended) The block deinterleaving apparatus of Claim ~~11~~ 11, wherein the values of α and $L^{(-x)}$ are set so that α is not equal to $L^{(-x)}$.

17. (Currently Amended) The block deinterleaving apparatus of Claim ~~11~~ 11, wherein the values of α, L, and M are set at 20, 8, and 203, respectively.

18. (Currently Amended) The block deinterleaving apparatus of Claim ~~11~~ 11, wherein the values of (L,M) are set at any of 72 possible values as follows:

~~L=96×X (X=1,2,4),~~ L=96xX, where X=1,2,4, and M=2,...,13,

or

M=2,...,13, and ~~L=96×X (X=1,2,4)~~ L-96xX, where X=1,2,4.

19. (Currently Amended) A block interleaving method for performing block interleaving of data ~~by~~ comprising:

generating addresses for writing and reading blocks, each block having (L×M) pieces of ~~data (L,M: integers, 2≤L,M)~~ data, wherein L and M are integers ≥2, as a unit to be interleaved, ~~in/from~~ into/out of a storage means to which (L×M) pieces of addresses are allocated, and controlling the storage means by using the generated addresses so that the storage means switches ~~the operation~~ between ~~the~~ data writing and ~~the~~ data ~~reading;~~ reading,

wherein, ~~α (integer, 2≤α)~~ α, an integer ≥2, is given as an address increment value REG to a block having a block number 0 and, thereafter, the increment value REG is multiplied by M every time the block number increments by 1 and thus obtained REG is used as an address increment value REG of the corresponding block, and when the address increment value REG exceeds ~~L×M-1,~~ (LxM)-1, the remainder over ~~L×M-1~~ (LxM)-1 is used as an increment value instead of the increment value REG to repeat the above-described processing, thereby performing a calculation equivalent to ~~"α×M**(b-x)mod(L×M-1)"~~ ~~(M**(b-x)~~ αxM**(b-x)mod((LxM)-1) where M**(b-x) indicates $M^{(b-x)}$, mod is the remainder, and x is an integer, such that, 0≤x≤b ~~0≤x≤b)~~ to obtain an address increment value of each block;

in the case where Ab(0) is set as an initial value of address in each block and, thereafter, the address increment value REG in ~~this~~ a specific block is successively summed to generate addresses Ab(1) to ~~Ab(n) (n: integer, 1≤n≤L×M-1) in this~~ Ab(n), wherein n is an integer such that 1≤n≤(LxM)-1 in said specific block, when the address exceeds ~~L×M-1,~~ (LxM)-1, the remainder over ~~L×M-1~~ (LxM)-1 is used as an address instead of the address to repeat the above-described processing, thereby generating addresses in each block; and

when calculating the address increment value, a decision is made as to whether the remainder is to be obtained or not is made by comparing the address increment value with the ~~L×M-1~~ (LxM)-1 using a first comparison means ~~and, at this time,~~ and the minimum value A which exceeds the ~~L×M-1~~ value (LxM)-1 and is included in the result of ~~multiplication~~ multiplication, is used as a comparison reference value instead of the ~~L×M-1.~~ value (LxM)-1.

20. (Currently Amended) The block interleaving method of Claim ~~19~~ 19, wherein said first comparison means employs, as a comparison reference value instead of the minimum value A exceeding the ~~L×M-1,~~ value (LxM)-1, a value B which satisfies ~~L×M-1<B<A~~ (LxM)-1<B<A and is selected so that the number of logic gates constituting the comparison means is minimized.

21. (Currently Amended) The block interleaving method of Claim ~~19~~ 19, wherein the values of α and ~~L×M-1~~ (LxM)-1 are set so that no common divisor exists between them.

22. (Currently Amended) The block interleaving method of Claim ~~19~~ 19, wherein the values of α and $M^{(-x)}$ are set so that α is not equal to $M^{(-x)}$.

23. (Currently Amended) The block interleaving method of Claim ~~19~~ 19, wherein the values of α, L, and M are set at 20, 8, and 203, respectively.

24. (Currently Amended) The block interleaving method of Claim ~~19~~ 19, wherein the values of (L,M) are set at any of 72 possible values as follows:

~~L=96×X (X=1,2,4),~~ L=96xX, where X=1,2,4, and M=2,...,13, or

M=2,...,13, where L=96×X ~~(X=1,2,4)~~ and X=1,2,4.

25. (Curerntly Amended) A block deinterleaving method for performing block deinterleaving of data ~~by~~ comprising:

generating addresses for writing and reading blocks, each block having (L×M) pieces of ~~data (L,M: integers, 2≤L,M)~~ data, wherein L and M are integers ≥2, as a unit to be deinterleaved, ~~in/from~~ into/from a storage means to which (L×M) pieces of addresses are allocated, and controlling the

storage means by using the generated addresses so that the storage means switches ~~the operation~~ between writing and reading of the data:

wherein, ~~α (integer, 2≤α)~~ α, an integer ≥2, is given as an address increment value REG to a block having a block number 0 and, thereafter, the increment value REG is multiplied by L every time the block number increments by 1 and thus obtained REG is used as an address increment value REG of the corresponding block, and when the address increment value REG exceeds ~~L×M-1,~~ (LxM)-1, the remainder over ~~L×M-1~~ (LxM)-1 is used as an increment value instead of the increment value REG to repeat the above-described processing, thereby performing a calculation equivalent to ~~"α×L**(b-x)mod(L×M-1)"~~ ~~(L**(b-x)~~ αxL**(b-x)mod ((LxM)-1) where L**(b-x) indicates $L^{(b-x)}$, mod is the remainder, and x is an integer, ~~0≤x≤b)~~ such that 0≤x≤b to obtain an address increment value of each block;

in the case where Ab(0) is set as an initial value of address in each block and, thereafter, the address increment value REG in ~~this~~ a specific block is successively summed to generate addresses Ab(1) to ~~Ab(n) (n: integer, 1≤n≤L×M-1)~~ Ab(n), wherein n is an integer such that 1≤n≤(LxM)-1 in ~~this~~ said specific block, when the address exceeds ~~L×M-1,~~ (LxM)-1,

the remainder over ~~L×M-1~~ (LxM)-1 is used as an address instead of the address to repeat the above-described processing, thereby generating addresses in each block; and

when calculating the address increment value, a decision is made as to whether the remainder is to be obtained or not is made by comparing the address increment value with the ~~L×M-1~~ (LxM)-1 using a first comparison means ~~and, at this time,~~ and the minimum value A which exceeds the ~~L×M-1~~ value (LxM)-1 and is included in the result of multiplication is used as a comparison reference value instead of the ~~L×M-1~~value (LxM)-1.

26. (Currently Amended) The block deinterleaving method of Claim ~~25~~ 25, wherein said first comparison means employs, as a comparison reference value instead of the minimum value A exceeding the ~~L×M-1,~~ value (LxM)-1, a value B which satisfies ~~L×M-1<B<A~~ (LxM)-1<B<A and is selected so that the number of logic gates constituting the comparison means is minimized.

27. (Currently Amended) The block deinterleaving method of Claim ~~25~~ 25, wherein the values of α and ~~L×M-1~~ (LxM)-1 are set so that no common divisor exists between them.

28. (Currently Amended) The block deinterleaving method of Claim ~~25~~ 25, wherein the values of α and $L^{(-x)}$ are set so that α is not equal to $L^{(-x)}$.

29. (Currently Amended) The block deinterleaving method of Claim ~~25~~ 25, wherein the values of α, L, and M are set at 20, 8, and 203, respectively.

30. (Currently Amended) The block deinterleaving method of Claim ~~25~~ 25, wherein the values of (L,M) are set at any of 72 possible values as follows:

~~L=96×X (X=1,2,4),~~ L=96xX wherein X=1,2,4, and M=2,...,13, or

M=2,...,13, where L=96×X ~~(X=1,2,4)~~ and X=1,2,4.